Exp.4 Exploratory Data Analysis

September 4, 2023

URK21CS1128AIM: To perform exploratory data analysis on the given dataset using various python libraries. DESCRIPTION:

[1]:
```
import pandas as pd

df = pd.read_csv('iris_EDA.csv')
df
```

[1]:
```
    sepallength  sepalwidth  petallength  petalwidth            class Name  \
0           5.1         3.5          1.4         0.2      Iris-setosa   F1
1           4.9         3.0          1.4         0.2      Iris-setosa   F2
2           4.7         3.2          1.3         0.2      Iris-setosa   F3
3           4.6         3.1          1.5         0.2      Iris-setosa   F4
4           5.0         3.6          1.4         0.2      Iris-setosa   F5
5           5.4         3.9          1.7         0.4      Iris-setosa   F6
6           4.6         3.4          1.4         0.3      Iris-setosa   F7
7           5.0         3.4          1.5         0.2      Iris-setosa   F8
8           7.0         3.2          4.7         1.4  Iris-versicolor   F9
9           6.4         3.2          4.5         1.5  Iris-versicolor  F10
10          6.9         3.1          4.9         1.5  Iris-versicolor  F11
11          5.5         2.3          4.0         1.3  Iris-versicolor  F12
12          6.5         2.8          4.6         1.5  Iris-versicolor  F13
13          5.7         2.8          4.5         1.3  Iris-versicolor  F14
14          6.3         3.3          4.7         1.6  Iris-versicolor  F15
15          4.9         2.4          3.3         1.0  Iris-versicolor  F16
16          6.3         3.3          6.0         2.5   Iris-virginica  F17
17          5.8         2.7          5.1         1.9   Iris-virginica  F18
18          7.1         3.0          5.9         2.1   Iris-virginica  F19
19          6.3         2.9          5.6         1.8   Iris-virginica  F20
20          6.5         3.0          5.8         2.2   Iris-virginica  F21
21          7.6         3.0          6.6         2.1   Iris-virginica  F22
22          4.9         2.5          4.5         NaN   Iris-virginica  F23
23          7.3         2.9          6.3         1.8   Iris-virginica  F24
24          7.3         2.9          6.3         1.8   Iris-virginica  F24

    Score  Color
0    12.0    Red
1     NaN   Blue
```

```
2    18.0  Orange
3    14.0  Purple
4    22.0     Red
5    27.0    Blue
6    24.0  Orange
7    23.0  Purple
8    16.0     Red
9    19.0    Blue
10   21.0  Orange
11   25.0  Purple
12   28.0     Red
13   29.0    Blue
14   11.0  Orange
15   30.0  Purple
16   12.0     Red
17   24.0    Blue
18   17.0  Orange
19   15.0  Purple
20   22.0     Red
21   27.0    Blue
22   25.0  Orange
23   21.0  Purple
24   21.0  Purple
```

Q1: Remove the irrelevant column 'Color' and display top 5 rows (use inplace=True)

[2]:

```python
print(1128)

df.drop('Color',axis=1,inplace=True)
print('Column dropped from dataframe permanently.')
```

```
1128
Column dropped from dataframe permanently.
```

[3]:

```python
print(1128)
df.shape
```

```
1128
```

[3]:

```
(25, 7)
```

Q2: Remove the duplicate rows and display the shape of the dataframe(use inplace=True).

[4]:

```python
print(1128)

df.drop_duplicates(keep='first',inplace=True)  #use 'subset' attribute for␣
 ↪dropping duplicates in individual columns
print('Dropped the duplicate rows.')
df.shape
```

```
1128
Dropped the duplicate rows.
```

```
[4]: (24, 7)
```

Q3: Rename the column 'class' to 'Category' and display top 5 rows (use inplace=True).

```
[5]: print(1128)
     df.rename(columns={'class':'Category'},inplace=True)
     print("Changed the column name 'class' to 'category' in the dataframe.")
     df.head()
```

```
1128
Changed the column name 'class' to 'category' in the dataframe.
```

```
[5]:    sepallength  sepalwidth  petallength  petalwidth     Category Name  Score
     0          5.1         3.5          1.4         0.2  Iris-setosa   F1   12.0
     1          4.9         3.0          1.4         0.2  Iris-setosa   F2    NaN
     2          4.7         3.2          1.3         0.2  Iris-setosa   F3   18.0
     3          4.6         3.1          1.5         0.2  Iris-setosa   F4   14.0
     4          5.0         3.6          1.4         0.2  Iris-setosa   F5   22.0
```

Q4:Drop the missing value row-wise and display the shape of dataframe (use inplace=False).

```
[6]: print(1128)
     df.dropna(axis=0,inplace=True)
     print('Dropped the rows with null/missing values in the dataframe.')
     df.shape
```

```
1128
Dropped the rows with null/missing values in the dataframe.
```

```
[6]: (22, 7)
```

Q5:Calculate the central tendency measures for 'Score' and display the same.

```
[7]: print(1128)

     print('Mean: ', df['Score'].mean())
     print('Median: ', df['Score'].median())
     print('Mode: ', df['Score'].mode())
```

```
1128
Mean:  20.772727272727273
Median:  21.5
Mode:  0    12.0
1    21.0
2    22.0
3    24.0
4    27.0
Name: Score, dtype: float64
```

Q6.Calculate the variability measures for ‘Score’ and display the same.

[8]:

```
print(1128)
x = df['Score'].min()
y = df['Score'].max()
print('Variability Measures for the column-Score: ')
print('Max: ',y)
print('Min: ',x)
print('Range:',(y-x))
print('Standard Deviation: ',df['Score'].std())
print('Variance: ',df['Score'].var())
```

```
1128
Variability Measures for the column-Score:
Max:  30.0
Min:  11.0
Range: 19.0
Standard Deviation:  5.797633492430693
Variance:  33.612554112554115
```

Q7.Calculate the IQR using quantile for ‘Score’ and display the same.

[9]:

```
print(1128)
Q1 = df['Score'].quantile(.25)
Q3 = df['Score'].quantile(.75)
print('IQR: ',(Q3-Q1))  #IQR formula=Q3-Q1
```

```
1128
IQR:  8.5
```

Q8. Calculate the z-score for ‘Score’ and display the same.

[10]:

```
print(1128)
#z-score = x-mean/SD
import  scipy.stats as stats

zscore = stats.zscore(df['Score'])
print('Z-score:',zscore)
```

```
1128
Z-score: 0    -1.548765
2    -0.489506
3    -1.195679
4     0.216667
5     1.099383
6     0.569753
7     0.393210
8    -0.842592
9    -0.312963
10    0.040123
```

```
11     0.746296
12     1.275926
13     1.452469
14    -1.725308
15     1.629012
16    -1.548765
17     0.569753
18    -0.666049
19    -1.019136
20     0.216667
21     1.099383
23     0.040123
Name: Score, dtype: float64
```

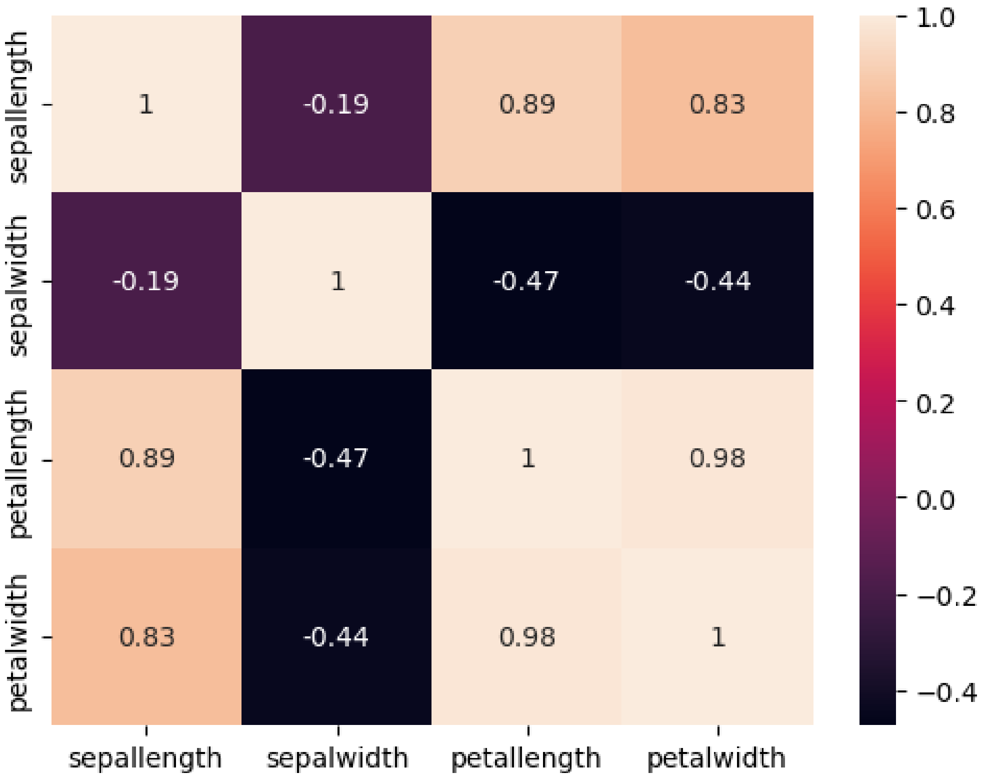
Q9: Plot the heatmap using the correlation (‘sepallength’, ‘sepalwidth’, ‘petallength’, ‘petalwidth’).

```
[11]: print(1128)
      import seaborn as sns

      t = df[['sepallength', 'sepalwidth', 'petallength', 'petalwidth']]
      c = t.corr()
      sns.heatmap(c, xticklabels = c.columns, yticklabels = c.columns, annot = True)
```

```
1128
```

```
[11]: <Axes: >
```



Q10:Add 2 rows at the end of the dataframe with the given values and display last 5 rows

{‘sepallength’:7.6,‘sepalwidth’:2.9,‘petallength’:5.3,‘petalwidth’:2.1,‘Category’:‘Iris-virginica’,‘Name’:‘F25’,‘Score’:80}

df2={‘sepallength’:4.6,‘sepalwidth’:1.3,‘petallength’:0.3,‘Category’:‘Iris-setosa’,‘Name’:‘F26’,‘Score’:85}

[19]:

```
print(1128)

df1 = {'sepallength':7.6,'sepalwidth':2.9,'petallength':5.3,'petalwidth':2.
 ↪1,'Category':'Iris-virginica','Name':'F25','Score':80}

df2 = {'sepallength':4.6,'sepalwidth':1.3,'petallength':0.3,'Category':
 ↪'Iris-setosa','Name':'F26','Score':85}

df = df.append(df1,ignore_index=True)
df = df.append(df2,ignore_index=True)
df.tail()
```

```
1128

/tmp/ipykernel_3676372/3128689531.py:7: FutureWarning: The frame.append method
is deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  df = df.append(df1,ignore_index=True)
/tmp/ipykernel_3676372/3128689531.py:8: FutureWarning: The frame.append method
is deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  df = df.append(df2,ignore_index=True)
```

[19]:

```
    sepallength  sepalwidth  petallength  petalwidth        Category Name  \
23          4.6         1.3          0.3    1.273913     Iris-setosa  F26
24          7.6         2.9          5.3    2.100000  Iris-virginica  F25
25          4.6         1.3          0.3         NaN     Iris-setosa  F26
26          7.6         2.9          5.3    2.100000  Iris-virginica  F25
27          4.6         1.3          0.3         NaN     Iris-setosa  F26

    Score
23   85.0
24   80.0
25   85.0
26   80.0
27   85.0
```

Q11: Replace NaN value in ‘petalwidth’ with mean petalwidth values and display last 5 rows.

[13]:
```
print(1128)
import numpy as np
m= df['petalwidth'].mean()
df.replace(to_replace=np.nan, value=m, inplace=True)
df.tail()
```

```
1128
```

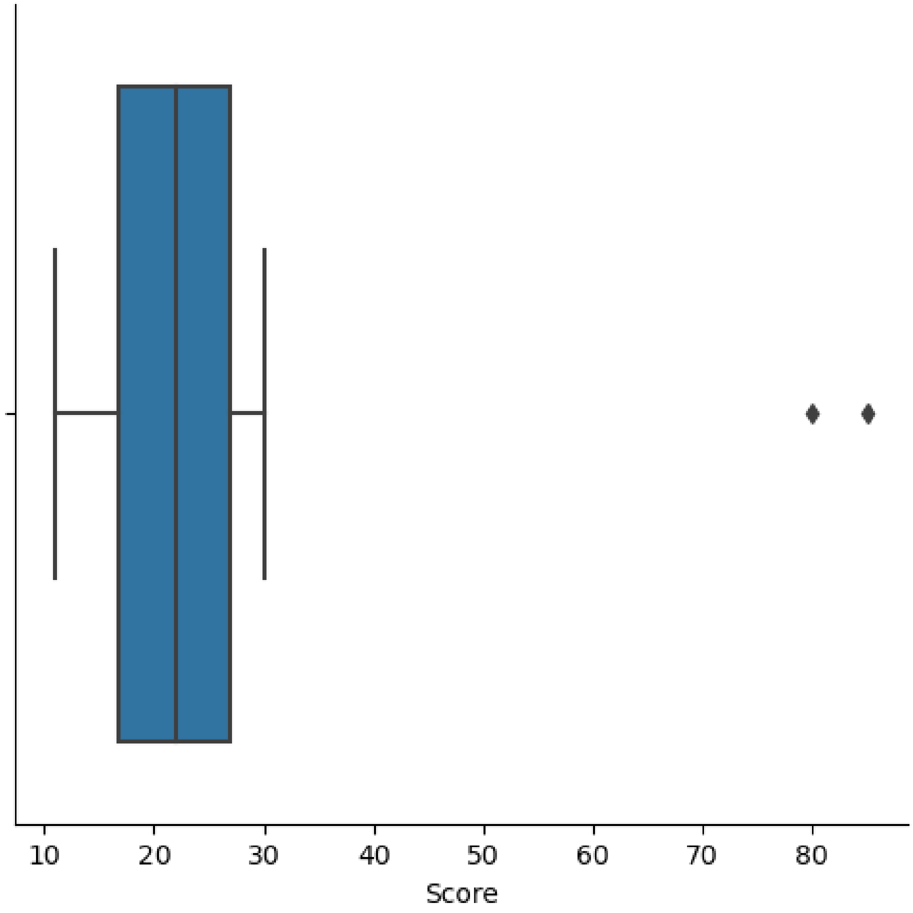
[13]:
```
    sepallength  sepalwidth  petallength  petalwidth        Category Name  \
19          6.5         3.0          5.8    2.200000  Iris-virginica  F21
20          7.6         3.0          6.6    2.100000  Iris-virginica  F22
21          7.3         2.9          6.3    1.800000  Iris-virginica  F24
22          7.6         2.9          5.3    2.100000  Iris-virginica  F25
23          4.6         1.3          0.3    1.273913     Iris-setosa  F26

    Score
19   22.0
20   27.0
21   21.0
22   80.0
23   85.0
```

Q12:Detect the outliers in ‘Score’ with boxplot.

[14]:
```
print(1128)
sns.catplot(x='Score', kind='box', data=df)
print('Mean: ', df['Score'].mean())
print('Standard Deviation: ',df['Score'].std())
print('Variance: ',df['Score'].var())
```

```
1128
Mean:  25.916666666666668
Standard Deviation:  18.301619473760205
Variance:  334.9492753623188
```

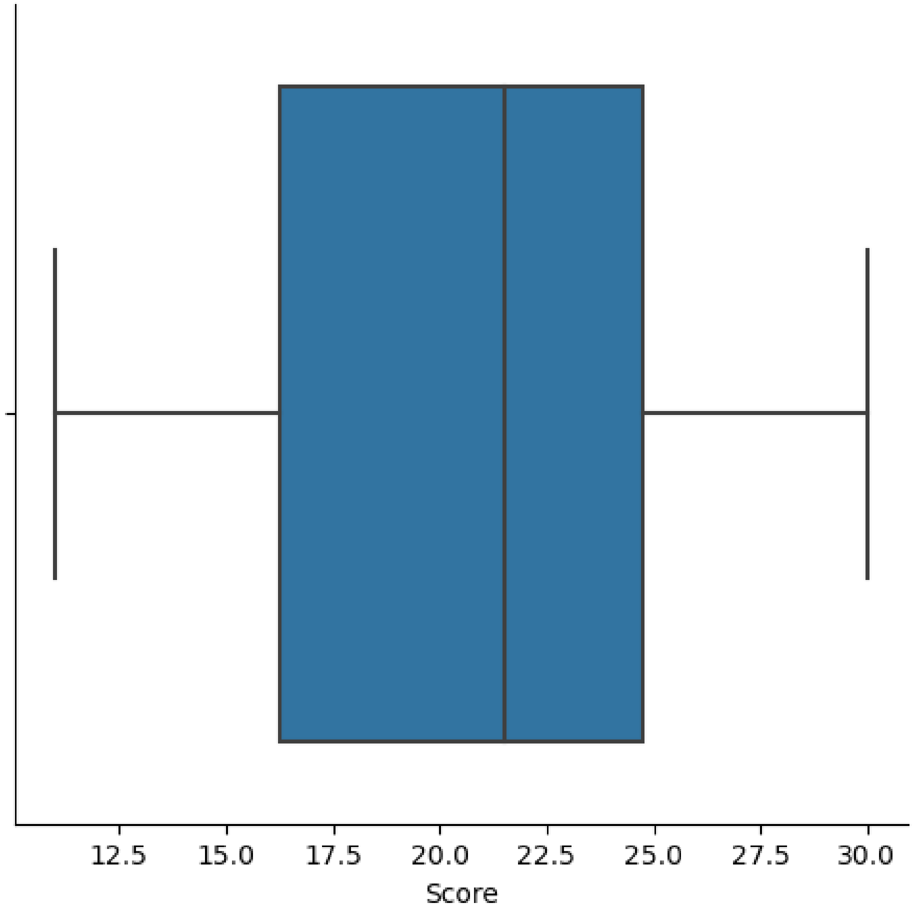


Q13:Remove the outliers using IQR and recalculate IQR in outlier removed ‘Score’ column and analyse with boxplot (Use df.copy()).

[15]:
```
print(1128)
Q1 = df['Score'].quantile(.25)
Q3 = df['Score'].quantile(.75)
IQR = Q3-Q1
print(Q1,Q3)
print('IQR: ',(Q3-Q1))
l = Q1-1.5*IQR
h = Q3+1.5*IQR
new_frame = df[(df['Score']>l) & (df['Score']<h)]
new_frame.shape
new_frame.tail()
sns.catplot(x='Score', kind='box', data=new_frame)
```

```
1128
16.75 27.0
IQR:  10.25
```

```
[15]: <seaborn.axisgrid.FacetGrid at 0x7f89126b5f40>
```



Q14: Remove the outliers using z-score and recalculate z-score in outlier removed ‘Score’ and analyse with boxplot (Use df.copy()).

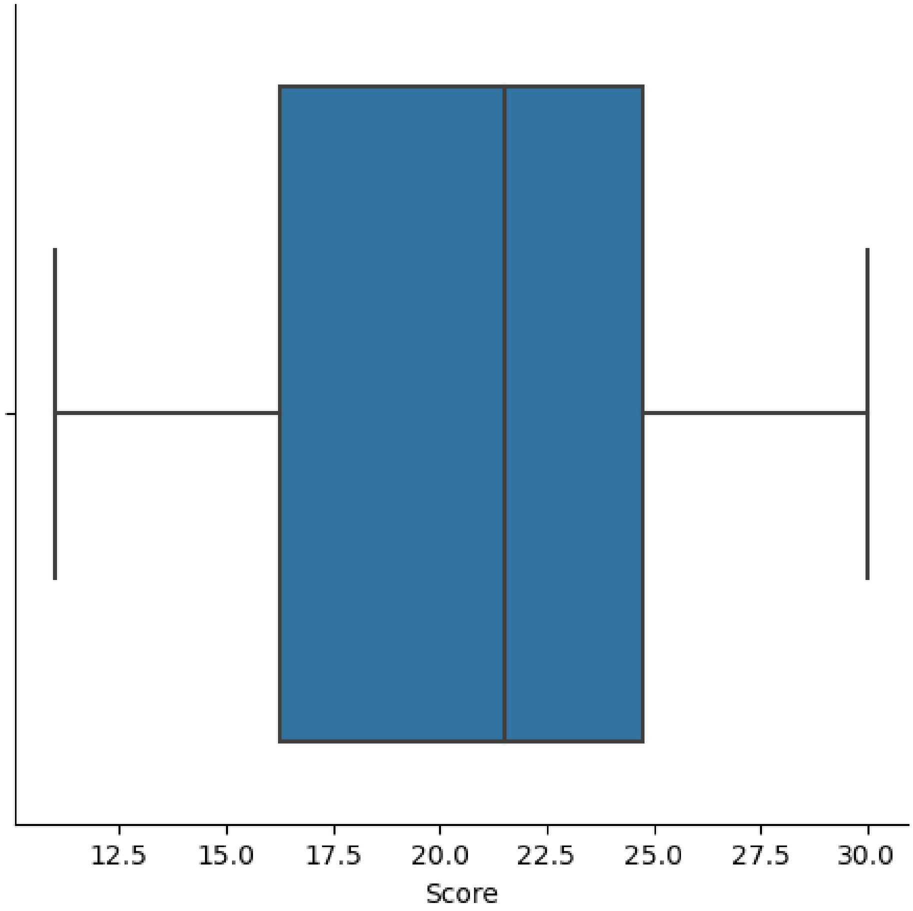
```
[16]: print(1128)
zscore = stats.zscore(df['Score'])
print('Z-score:',zscore)

filtered = (zscore<3)
new_df2 = df[filtered]
new_df2.tail()
sns.catplot(x='Score', kind='box', data=new_df2)
```

```
1128
Z-score: 0    -0.776761
1    -0.441870
2    -0.665131
3    -0.218609
```

```
4     0.060466
5    -0.106979
6    -0.162794
7    -0.553500
8    -0.386055
9    -0.274425
10   -0.051164
11    0.116282
12    0.172097
13   -0.832576
14    0.227912
15   -0.776761
16   -0.106979
17   -0.497685
18   -0.609316
19   -0.218609
20    0.060466
21   -0.274425
22    3.018670
23    3.297746
Name: Score, dtype: float64
```

```
[16]: <seaborn.axisgrid.FacetGrid at 0x7f89126c7a60>
```



Q15:Drop the last two rows added in the dataframe.

```
[2]: #15 Drop the last two rows added in the dataframe
print('URK21CS1128')
df = df.drop([22,23])
df.shape
print(df.to_string())
```

```
URK21CS1128
    sepallength  sepalwidth  petallength  petalwidth            class Name
Score   Color
0           5.1         3.5          1.4         0.2      Iris-setosa   F1
12.0     Red
1           4.9         3.0          1.4         0.2      Iris-setosa   F2
NaN    Blue
2           4.7         3.2          1.3         0.2      Iris-setosa   F3
18.0  Orange
3           4.6         3.1          1.5         0.2      Iris-setosa   F4
14.0  Purple
```

```
4            5.0          3.6           1.4          0.2       Iris-setosa   F5
22.0     Red
5            5.4          3.9           1.7          0.4       Iris-setosa   F6
27.0    Blue
6            4.6          3.4           1.4          0.3       Iris-setosa   F7
24.0  Orange
7            5.0          3.4           1.5          0.2       Iris-setosa   F8
23.0  Purple
8            7.0          3.2           4.7          1.4   Iris-versicolor   F9
16.0     Red
9            6.4          3.2           4.5          1.5   Iris-versicolor  F10
19.0    Blue
10           6.9          3.1           4.9          1.5   Iris-versicolor  F11
21.0  Orange
11           5.5          2.3           4.0          1.3   Iris-versicolor  F12
25.0  Purple
12           6.5          2.8           4.6          1.5   Iris-versicolor  F13
28.0     Red
13           5.7          2.8           4.5          1.3   Iris-versicolor  F14
29.0    Blue
14           6.3          3.3           4.7          1.6   Iris-versicolor  F15
11.0  Orange
15           4.9          2.4           3.3          1.0   Iris-versicolor  F16
30.0  Purple
16           6.3          3.3           6.0          2.5    Iris-virginica  F17
12.0     Red
17           5.8          2.7           5.1          1.9    Iris-virginica  F18
24.0    Blue
18           7.1          3.0           5.9          2.1    Iris-virginica  F19
17.0  Orange
19           6.3          2.9           5.6          1.8    Iris-virginica  F20
15.0  Purple
20           6.5          3.0           5.8          2.2    Iris-virginica  F21
22.0     Red
21           7.6          3.0           6.6          2.1    Iris-virginica  F22
27.0    Blue
24           7.3          2.9           6.3          1.8    Iris-virginica  F24
21.0  Purple
```

Result:

```
[ ]: The basic functionalities of data visualization using python were executed␣
       ↪successfully.
```

```
[ ]:
```

```
[ ]:
```